VBA: Build Simple LRS Routes Using Attribute Table To Calibrate

Contributed by Bert Granberg
01, Jan. 2010

This script is for building single part calibrated route features where the calibration values are housed in a standalone table.

Specifically it was designed for UDOT ramps and collectors where all component street features are oriented in the direction of travel (ascending route coordinates).This could be potentially be modified for use with Federal Aid Eligible Routes.

After building the routes, use the script checkForDoubleZeroMCoordinates (included below) to check the second vertex for 0.000 measure values. These need to be fixed to avoid issues with loading the routes into Oracle Spatial at UDOT.

!!!!!!!!

ALSO NEEDS THESE FUNCTIONS (from: http://gis.utah.gov/code-visual-basic/vba-build-simple-calibrated-routes) :

createRouteFeatureClass()
buildRoutePartPolyline()

!!!!!!!

```
Public Sub BuildRampRoutes()

    Dim outPath As String
    Dim calibrationTableIndex As Integer
    Dim roadsLayerIndex As Integer
    Dim pCalibrationEndValueFieldName As String
    Dim calibrationTableRouteFieldName As String
    Dim outReportPath As String
    Dim rampSQLWhereClauseStr As String
    Dim roadsLayerRouteFieldName As String

    '***************************
    '***** SET THESE PARAMETERS

    'the file geodatabase where the new route/polyline-m feature class will be created
    outPath = "c:/UDOTLRS/LRSBert.gdb"

    'the position of the CALIBRATION Table in the MXD, zero is the first table
    calibrationTableIndex = 0

    'the position of the ROAD CENTERLINE layer in the MXD, zero is the first layer
    roadsLayerIndex = 3

    'the name of the field containing the routename in the roadslayers
    roadsLayerRouteFieldName = "DOT_RTNAME"

    'The field name containing the measure value in the CALIBRATION LAYER
    pCalibrationEndValueFieldName = "END_ACCUMU"

    'The field name containing the full route-part name value in the CALIBRATION LAYER
    'Values in this fields must match the format:
    'concatenation of RT_NAME & "_" & RT_PART from the road centerline data
    calibrationTableRouteFieldName = "ROUTE_NAME"

    'Location for error report text file
    outReportPath = "C:\UDOTLRS\"

    'Queryfilter where clause that defines a subset of street features that are ramps
```

```
    rampSQLWhereClauseStr = "len(DOT_RTNAME) = 11"
    '***** END PARAMETERS
    '*************************

    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pTC As ITableCollection
    Dim pCalibrationTable As ITable
    Dim pRoadsLayer As IFeatureLayer
    Dim pRoadsFC As IFeatureClass
    Dim pOutWS As IFeatureWorkspace
    Dim pOutFields As IFields
    Dim pOutFC As IFeatureClass
    Dim pGDS As IGeoDataset
    Dim pOutSR As ISpatialReference
    Dim datestamp As String

    Set pMxDoc = ThisDocument
    Set pMap = pMxDoc.FocusMap
    Set pTC = pMap 'QI
    Set pCalibrationTable = pTC.Table(calibrationTableIndex)
    Set pRoadsLayer = pMap.Layer(roadsLayerIndex)
    Set pRoadsFC = pRoadsLayer.FeatureClass
    Set pGDS = pRoadsFC


    Set pOutSR = pGDS.SpatialReference

    datestamp = Format(Now, "yyyymmddhhmmss")
    'Open outReportPath & "rtreport_" & datestamp & ".txt" For Output As #1

    Set pOutWS = openFGDBWS(outPath)
    Set pOutFields = createRampFields(esriGeometryPolyline, pOutSR, True)
    Set pOutFC = createRouteFeatureClass(pOutWS, "RampsAndColls" & datestamp, esriFTSimple, esriGeometryPolyline,
pOutFields)

    Dim csvRoutePartList As String
    Dim routePartList() As String
    Dim p As Long
    Dim routePartQStr As String
    Dim pRoutePartPolyline As IPolyline
    Dim rampQF As IQueryFilter
    Dim point1M, point2m As Double
    Dim pMAware As IMAware
    Dim pMSegmentation As IMSegmentation3
    Dim pOutFeature As IFeature

    Set rampQF = New QueryFilter
    rampQF.WhereClause = rampSQLWhereClauseStr

    'Build Route Part List
    csvRoutePartList = getUniqueValues(pRoadsLayer, roadsLayerRouteFieldName, rampQF)
    routePartList = Split(csvRoutePartList, ",")

    For p = 0 To UBound(routePartList)
       'Debug.Print routePartList(p)

       routePartQStr = "DOT_RTNAME = '" & routePartList(p) & "'"
       Set pRoutePartPolyline = buildRoutePartPolyline(routePartQStr, pRoadsFC)

       If Not pRoutePartPolyline.IsEmpty Then

            point1M = 0

            Dim pCalQF As IQueryFilter
```

```
            Dim pCalCursor As ICursor
            Dim pCalRow As IRow

            Set pCalQF = New QueryFilter
            pCalQF.WhereClause = calibrationTableRouteFieldName & " = '" & routePartList(p) & "'"
            Set pCalCursor = pCalibrationTable.Search(pCalQF, True)
            Set pCalRow = pCalCursor.NextRow
            If Not pCalRow Is Nothing Then
               If pCalRow.value(pCalibrationTable.FindField(pCalibrationEndValueFieldName)) = 0 Then
                  point2m = pRoutePartPolyline.Length / 1609.344
               Else
                  point2m = pCalRow.value(pCalibrationTable.FindField(pCalibrationEndValueFieldName))
               End If
            Else
               point2m = pRoutePartPolyline.Length / 1609.344
            End If

            Set pMAware = pRoutePartPolyline
            pMAware.MAware = True
            Set pMSegmentation = pRoutePartPolyline
            pMSegmentation.SetAndInterpolateMsBetween point1M, point2m
'
'         End If
         Set pOutFeature = pOutFC.CreateFeature

         Debug.Print routePartList(p)
         Set pOutFeature.Shape = pRoutePartPolyline
         pOutFeature.value(pOutFeature.Fields.FindField("LABEL")) = routePartList(p)
         pOutFeature.value(pOutFeature.Fields.FindField("RT_NAME")) = Left(routePartList(p), 4)
         pOutFeature.value(pOutFeature.Fields.FindField("RT_DIR")) = Mid(routePartList(p), 5, 1)
         pOutFeature.value(pOutFeature.Fields.FindField("INTERCHANGE")) = Mid(routePartList(p), 7, 3)
         pOutFeature.value(pOutFeature.Fields.FindField("RAMP")) = Mid(routePartList(p), 10, 2)

         If Not pCalRow Is Nothing Then

            pOutFeature.value(pOutFeature.Fields.FindField("UDOT_NAME")) =
pCalRow.value(pCalibrationTable.FindField("COMMON_NAM"))
            pOutFeature.value(pOutFeature.Fields.FindField("UDOT_DESC")) =
pCalRow.value(pCalibrationTable.FindField("ROUTE_DESC"))
            pOutFeature.value(pOutFeature.Fields.FindField("UDOT_START")) =
pCalRow.value(pCalibrationTable.FindField("ROUTE_STAR"))
            pOutFeature.value(pOutFeature.Fields.FindField("UDOT_END")) =
pCalRow.value(pCalibrationTable.FindField("ROUTE_END_"))

         End If

         pOutFeature.value(pOutFeature.Fields.FindField("EFF_DATE")) = Now
         pOutFeature.Store

      End If
   Next p


   Close #1

End Sub
Public Function createRampFields(geomType As Long, pSR As ISpatialReference, _
                        hasM As Boolean) As IFields
   Dim pField As IField
   Dim pFields As IFields
   Dim pFieldEdit As IFieldEdit
   Dim pFieldsEdit As IFieldsEdit
   Dim hasmcoord As Boolean
```

```
'Create new Fields collection
Set pFields = New Fields
Set pFieldsEdit = pFields
'pFieldsEdit.FieldCount = 1

''
'' create the geometry field
''
Dim pGeomDef As IGeometryDef
Set pGeomDef = New GeometryDef
Dim pGeomDefEdit As IGeometryDefEdit
Set pGeomDefEdit = pGeomDef

' assign the spatial reference
'Dim pSR As ISpatialReference
If pSR Is Nothing Then
    Set pSR = New UnknownCoordinateSystem
    pSR.SetFalseOriginAndUnits 0, 0, 100
End If

pSR.SetMFalseOriginAndUnits -100000, 1000

If Not hasM Then
    hasmcoord = False
Else
    hasmcoord = True
End If

'' assign the geometry definiton properties.
With pGeomDefEdit
  .GeometryType = geomType
  .GridCount = 1
  .GridSize(0) = 560000
  .AvgNumPoints = 200
  .hasM = hasmcoord
  .HasZ = False
  Set .SpatialReference = pSR
End With

Set pField = New Field
Set pFieldEdit = pField

pFieldEdit.Name = "Shape"
pFieldEdit.Type = esriFieldTypeGeometry
Set pFieldEdit.GeometryDef = pGeomDef
pFieldsEdit.AddField pField


'Create Object ID Field
Set pField = New Field
Set pFieldEdit = pField

With pFieldEdit
    .Name = "OBJECTID"
    .AliasName = "FID"
    .Type = esriFieldTypeOID
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
    .Length = 11
    .Name = "LABEL"
    .Type = esriFieldTypeString
```

```
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 4
   .Name = "RT_NAME"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 1
   .Name = "RT_DIR"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 3
   .Name = "INTERCHANGE"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 2
   .Name = "RAMP"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Name = "RT_DIR_ID"
   .Type = esriFieldTypeInteger
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 30
   .Name = "AGRC_NAME"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField


Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 80
   .Name = "UDOT_NAME"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField
```

```
Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 80
   .Name = "UDOT_DESC"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 80
   .Name = "UDOT_START"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 80
   .Name = "UDOT_END"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Name = "EFF_DATE"
   .Type = esriFieldTypeDate
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Name = "DEP_DATE"
   .Type = esriFieldTypeDate
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 100
   .Name = "EFF_NOTES"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pField = New Field
Set pFieldEdit = pField
With pFieldEdit
   .Length = 100
   .Name = "DEP_NOTES"
   .Type = esriFieldTypeString
End With
pFieldsEdit.AddField pField

Set pFields = pFieldsEdit

Set createRampFields = pFields
```

```
End Function

Public Sub checkForDoubleZeroMCoordinates()
    Dim datestamp As String
    Dim pMxDoc As IMxDocument
    Dim pMap As IMap
    Dim pRouteLayer As IFeatureLayer
    Dim pRoutePartsLayer As IFeatureLayer
    Dim pRouteFCursor As IFeatureCursor
    Dim pRouteFeature As IFeature
    Dim pRoutePolyline As IPolyline
    Dim routeLabelStr As String
    Dim pQF As IQueryFilter

    Set pMxDoc = ThisDocument


    Set pMap = pMxDoc.FocusMap
    Set pRouteLayer = pMap.Layer(2)

    Set pQF = New QueryFilter
    Set pRouteFCursor = pRouteLayer.Search(pQF, True)
    Set pRouteFeature = pRouteFCursor.NextFeature

    Dim lastMCoordinate As Double
    Dim currMCoordinate As Double
    Dim pRoutePtCollection As IPointCollection
    Dim pPoint As IPoint
    Dim p As Long
    Dim pEnvelope As IEnvelope

    Do Until pRouteFeature Is Nothing

        Set pRoutePtCollection = pRouteFeature.Shape
        p = 1
        Set pPoint = pRoutePtCollection.Point(p)
        currMCoordinate = pPoint.M

            If currMCoordinate = 0 And Not pPoint.IsEmpty Then
                Set pEnvelope = pRouteFeature.Extent
                pEnvelope.Width = 5
                pEnvelope.Height = 5
                pEnvelope.CenterAt pPoint
                pMxDoc.ActiveView.Extent = pEnvelope
                pMxDoc.ActiveView.Refresh
                Debug.Print pRouteFeature.value(pRouteFeature.Fields.FindField("LABEL"))
                Debug.Print " "
            End If
        Set pRouteFeature = pRouteFCursor.NextFeature
    Loop
    Close #1
End Sub
```